AirPlay and External Displays in iOS Apps

Session 406

Jim Batson and Josh Shaffer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

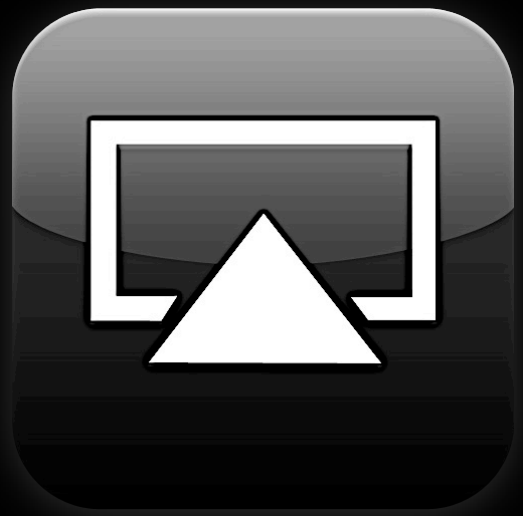
What You Will Learn

- Making a great AirPlay experience
- Getting the most out of video
- Taking advantage of multiple displays

2nd Display

Photos

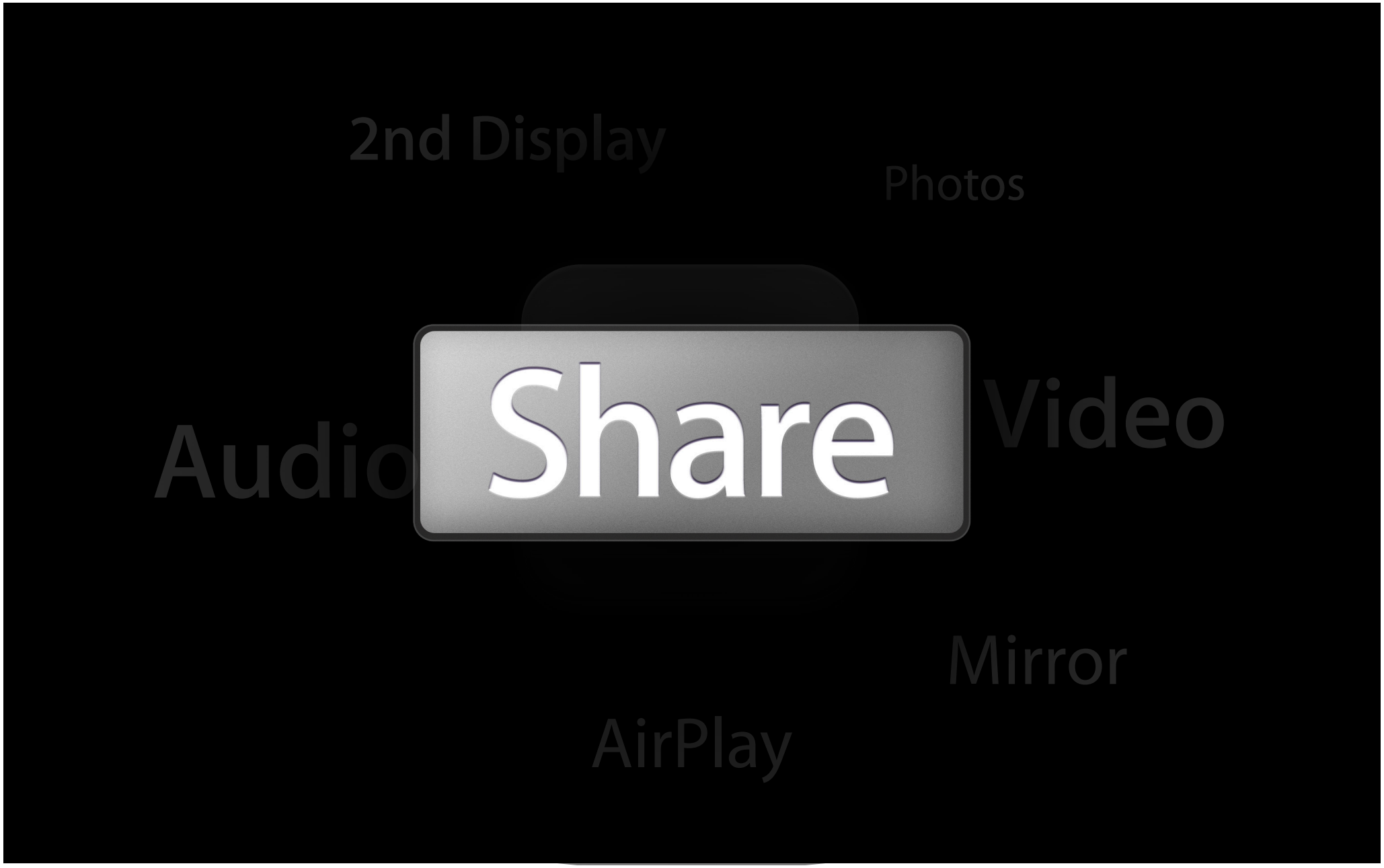
Audio



Video

Mirror

AirPlay


2nd Display
Photos
Audio
Share
Video
Mirror
AirPlay

Audio

WWDC
2011
Top Movies
Movies
TV Shows
Internet
Computers
Settings
Top Movies
Genres
Search
In Theaters
tv

Audio
WWDC
2011
airborne
WWDC
2011
1:28
-1:47

Audio
WWDC
2011
airborne
WWDC
2011
1:28
-1:47

Video
AirPlay
This video is playing on "Great Room Apple TV".

Video
Apple Store
For everything Dad does,
the gift that does everything.

Wired
iPad
9:41 AM
Messages
Calendar
Notes
Reminders
Maps
YouTube
Videos
Contacts
Game Center
iTunes
App Store
Newsstand
FaceTime
Camera
Photo Booth
Settings
Safari
Mail
Photos
Music

Screen
iPad
9:41 AM
Messages
Calendar
Notes
Reminders
Maps
YouTube
Videos
Contacts
Game Center
iTunes
App Store
Newsstand
FaceTime
Camera
Photo Booth
Settings
Safari
Mail
Photos
Music

Screen
Presentations
Undo
Presentation
2011

Screen
00:01:52
current
next
2011
2011
2011

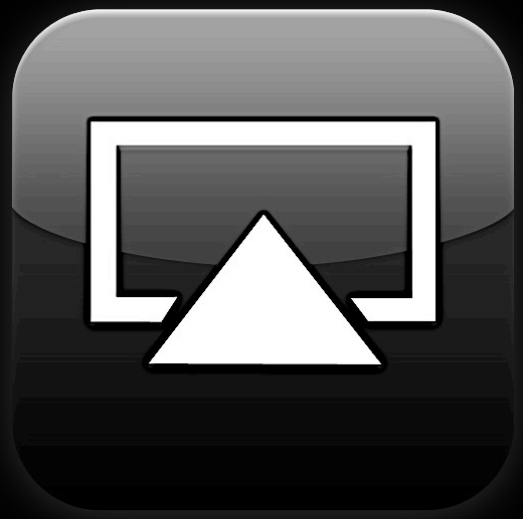
Screen
QUICK RACE
3 LAPS
MCKINLEY CIRCUIT
POSITION
12/16
LAP
1/3
SPEED
117 MPH
TIME
00:12.450
16 / 16
2 / 3
74 MPH
00:31.367

MediaPlayer

Mirroring

Routing

HTTP Live Streaming



UIScreen

Multitasking

Remote Control

Second Display

NowPlayingInfoCenter

AVFoundation

What You Will Learn

- Making a great AirPlay experience
- Getting the most out of video
- Taking advantage of multiple displays

What You Will Learn

- Making a great AirPlay experience
  - Understand media routing
  - Add AirPlay UI
  - Show what is playing
  - Accept remote controls
  - Master multitasking

Media Routing

Media Routes

Last in wins!

Audio Routing Behavior

Different uses

- Application Audio
  - Ambient sounds, incidental sounds, media playback
- System Sounds
  - Key clicks, alerts, notifications

Audio Routing Behavior

System sounds and AirPlay

AVAudioPlayer

- Create from file URL

```
AVAudioPlayer *player =
    [AVAudioPlayer alloc] initWithContentsOfURL:url withError:&error];
```

- Control playback

```
[player prepareToPlay/play/pause/stop];
```

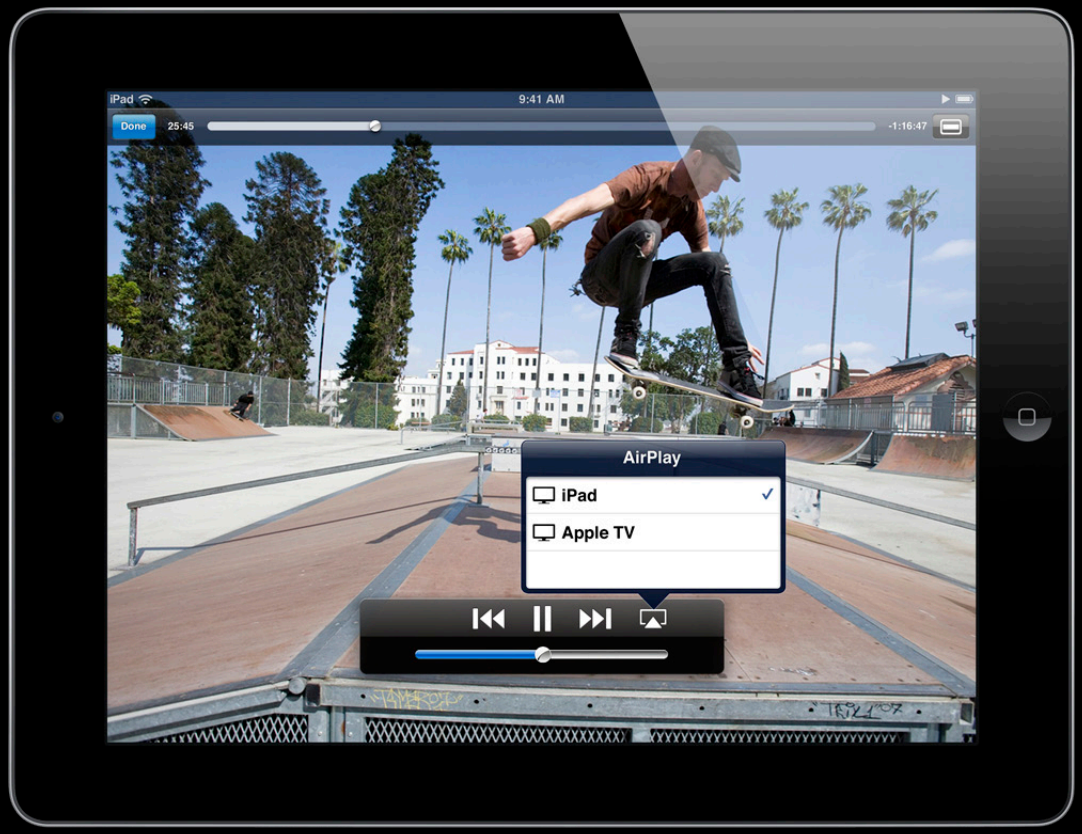
AirPlay Picker

System AirPlay Picker

AirPlay Picker

Help the user

- Media applications
  - Easy access for user
  - Helpful indicator



AirPlay Picker

MPVolumeView

- Volume and Picker

```
MPVolumeView *volumeView = [[MPVolumeView alloc] init];
[view addSubview:volumeView];
```

- Picker only

```
MPVolumeView *volumeView = [[MPVolumeView alloc] init];
[volumeView setShowsVolumeSlider:NO];
[volumeView sizeToFit];
[view addSubview:volumeView];
```

Media
Info Display

What's the Name of That Song?

What's the Name of That Song?

MPNowPlayingInfoCenter

What's the Name of That Song?

MPNowPlayingInfoCenter

- Part of MediaPlayer Framework
- Works with any playback framework
  - MediaPlayer, AVFoundation, AudioQueue, etc

```
[[MPNowPlayingInfoCenter defaultCenter] setNowPlayingInfo:dictionary];
```

What's the Name of That Song?

MPNowPlayingInfoCenter

- Song name
  - `MPMediaItemPropertyTitle`
- Artist
  - `MPMediaItemPropertyArtist`
- Album
  - `MPMediaItemPropertyAlbumTitle`
- Artwork
  - `MPMediaItemPropertyArtwork`
- More…

What's the Name of That Song?

MPNowPlayingInfoCenter

- Playback Rate
  - `MPNowPlayingInfoPropertyPlaybackRate`
- Duration
  - `MPMediaItemPropertyPlaybackDuration`
- Elapsed Time
  - `MPNowPlayingInfoPropertyElapsedPlaybackTime`
- More…

Remote Control Events

Take Control

Receiving remote control events

Take Control

Receiving remote control events

- Begin receiving remote controls

```
– (BOOL)canBecomeFirstResponder { return YES; }
– (void)viewDidAppear:(BOOL)animated {
  [super viewDidAppear:animated];
  [[UIApplication sharedApplication] beginReceivingRemoteControlEvents];
  [self becomeFirstResponder];
}
```

- End receiving remote controls

```
– (void)viewWillDisappear:(BOOL)animated {
    [super viewWillDisappear:animated];
    [[UIApplication sharedApplication] endReceivingRemoteControlEvents];
    [self resignFirstResponder];
}
```

Take Control

Responding to remote control events

- Override UIResponder method

```
- (void) remoteControlReceivedWithEvent: (UIEvent *) receivedEvent {
    if (receivedEvent.type == UIEventTypeRemoteControl) {
        switch (receivedEvent.subtype) {
            case UIEventSubtypeRemoteControlTogglePlayPause:
                [self playOrStop: nil];
                break;
            case UIEventSubtypeRemoteControlPreviousTrack:
                [self previousTrack: nil];
                break;
            case UIEventSubtypeRemoteControlNextTrack:
                [self nextTrack: nil];
                break;
            // ... More events see UIEvent.h
```

Multitasking and Media

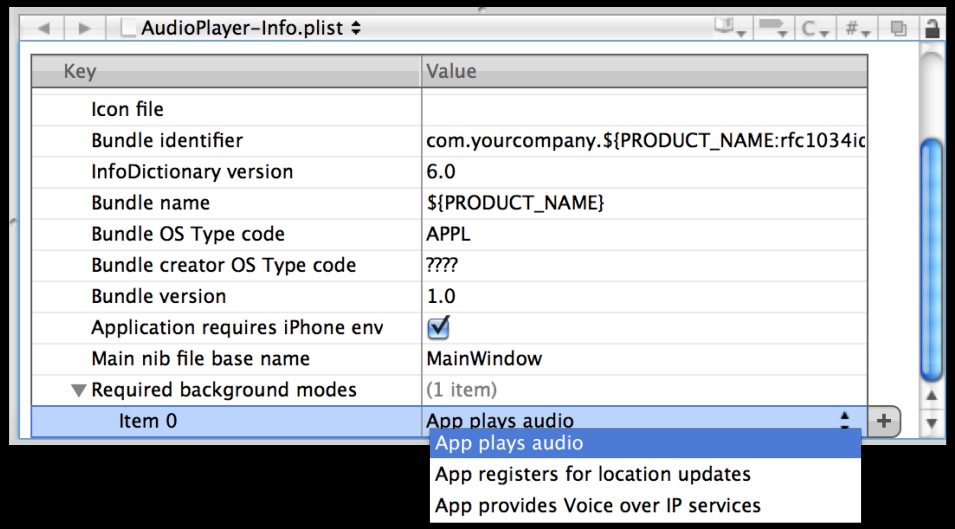
Multitasking and Media

- Play audio in background
- AirPlay video in background

Multitasking and Media

Background modes

- Info.plist
  - Required background modes
    - App plays audio



Multitasking and Media

Implementing audio behaviors

- AVAudioSession
  - define and handle audio behavior

```
[session setCategory:AVAudioSessionCategoryPlayback error:&error]
```

Audio Session Management for iOS

Marina
Wednesday 11:30AM

Multitasking and Media

Complete experience

- Combine
  - Now Playing info
  - Remote control events
  - Background media

What You Will Learn

- Making a great AirPlay experience
- Getting the most out of video
- Taking advantage of multiple displays

What You Will Learn

- Making a great AirPlay experience
- Getting the most out of video
  - Understand available video APIs
  - Determine when to allow AirPlay video
  - Show users where video is playing
  - Provide the best fullscreen experience

Video APIs

Video Content

Local

Internet

Progressive Download

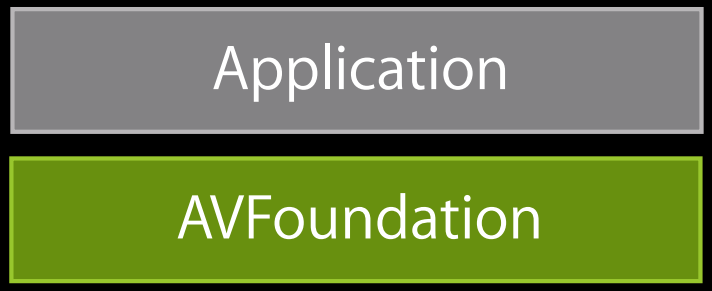
HTTP Live Streaming

HTTP Live Streaming

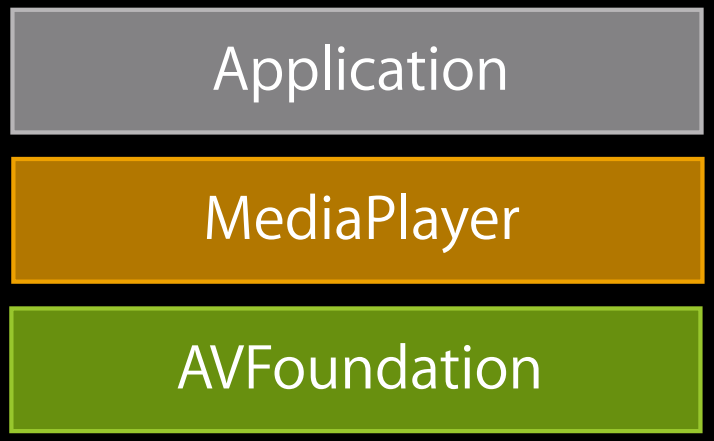
Encrypted streams and AirPlay

- Encrypted stream will work with AirPlay video
  - If you follow the recommendations for encrypted streams!
- Recommendation
  - Use NSURLProtocol to provide keys to media player
  - Use HTTPS and session cookie to provide keys

Video APIs



Video APIs



Allowing AirPlay Video

Allowing AirPlay Video

When to allow

Allow	Disable
▪ Primary playback experience ▪ High quality video	▪ Incidental videos ▪ Audio played with AVPlayer

In iOS 5, AirPlay video is allowed by default

Allowing AirPlay Video

Default behavior

	iOS 4.3	iOS 5
MediaPlayer	Disabled	Allowed
AVFoundation	N/A	Allowed
UIWebView	N/A	Allowed
Webpages	Disabled	Allowed

Explicitly allow/disable when targeting multiple iOS releases

Allowing AirPlay Video

Applications

- MPMoviePlayerController

```
mpPlayer.allowsAirPlay = YES;
```

- AVPlayer

```
avPlayer.allowsAirPlayVideo = YES;
```

- UIWebView

```
webView.mediaPlaybackAllowsAirPlay = YES;
```

Use "NO" to disable AirPlay video

Allowing AirPlay Video

Webpages

- <video>

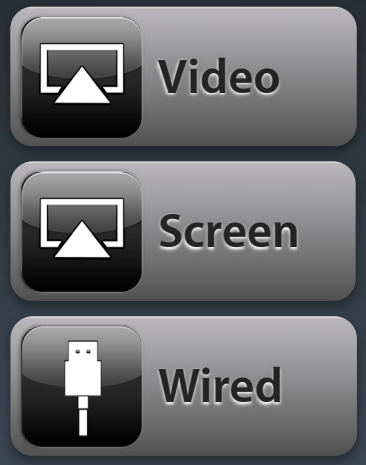
```
<video x-webkit-airplay="allow" poster="trailer_poster.png"
      src="trailer_video.m4v" controls>...</video>
```

- <embed>

```
<embed airplay="allow" src="movie.mov" width=320 height=240
      mime-type="video/quicktime">...</embed>
```

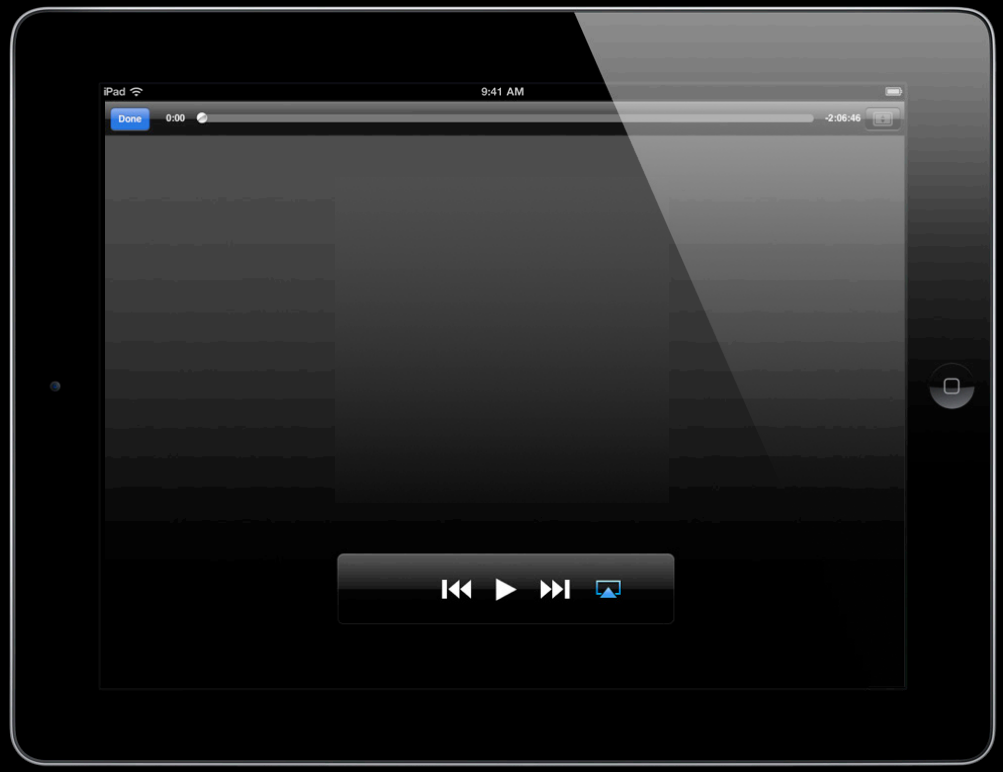
Use "deny" to disable AirPlay video

Where Is Video Playing?



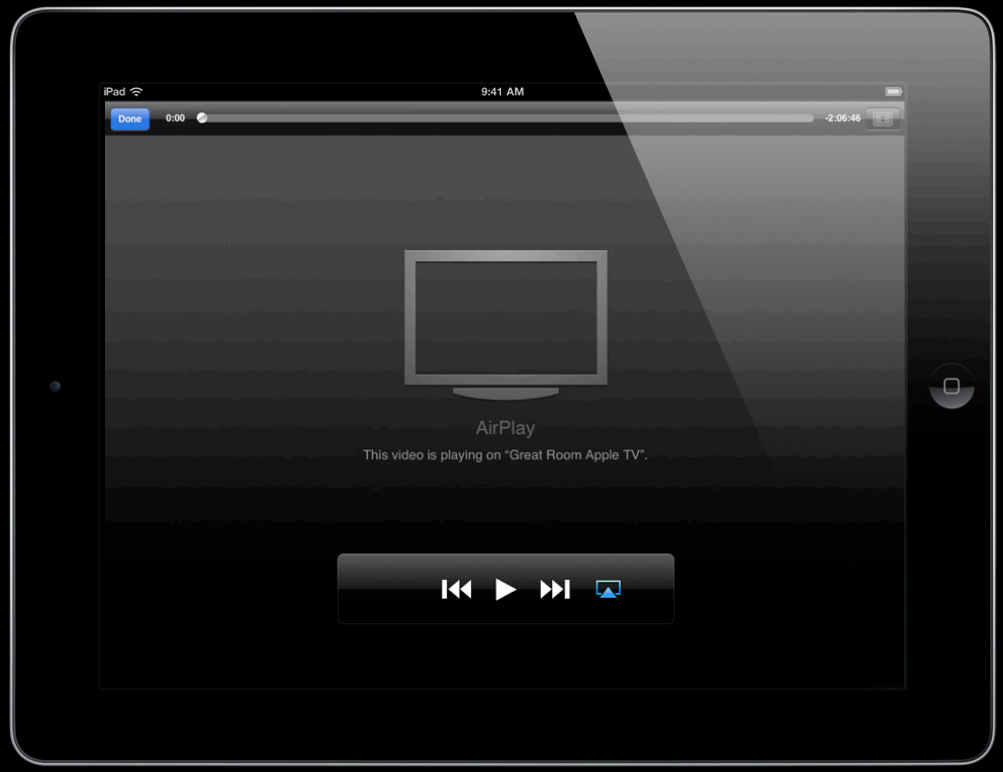
Where Is Video Playing?

Don’t leave user in the dark



Where Is Video Playing?

Let them know where to look



Where Is Video Playing?

MPMoviePlayerController

UI Automatically Displayed

Where Is Video Playing?

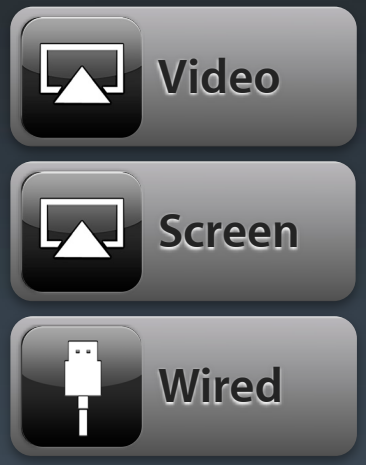
AVPlayer

- No "Video is playing on..." message
- Instead, you have to do this yourself

```
if (player.airPlayVideoActive == YES)
    [myView displayHelpfulMessage];
```

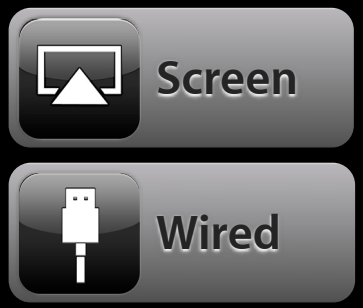
- Listen to changes through Key-Value Observing

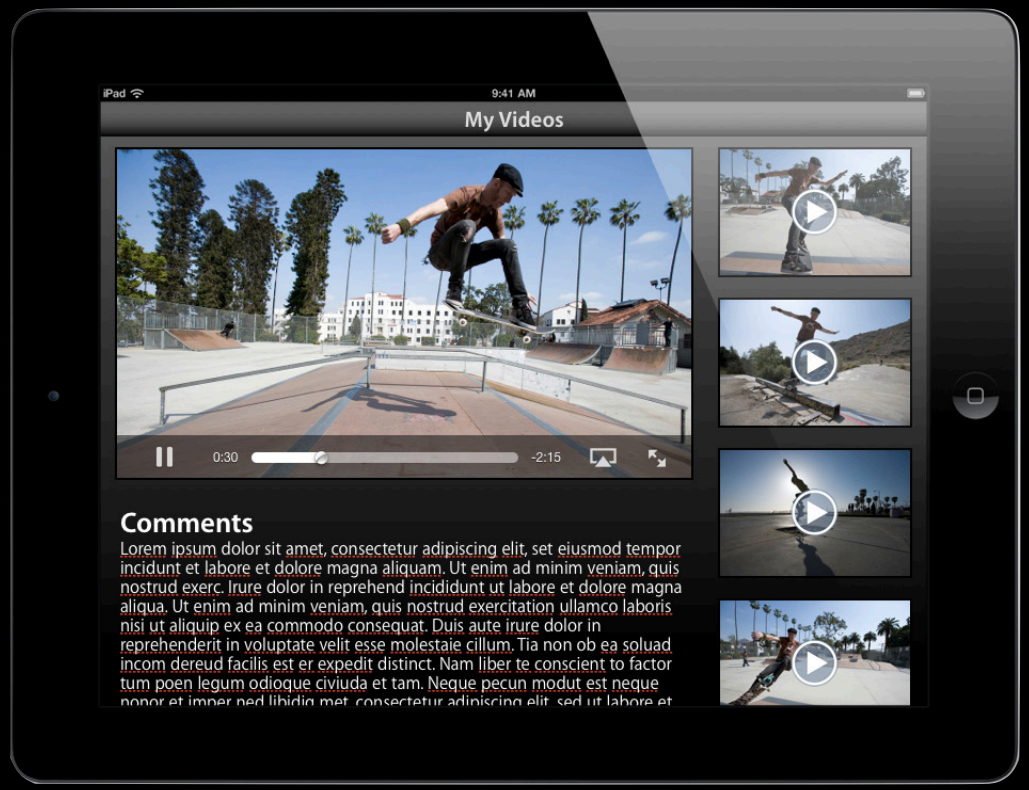
Fullscreen Video Experience



Fullscreen Video Experience

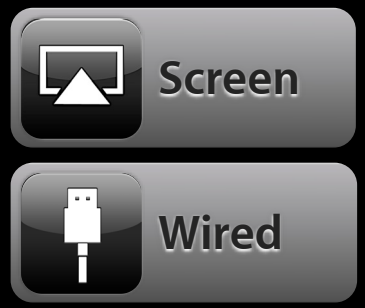
Recommended external screen behavior





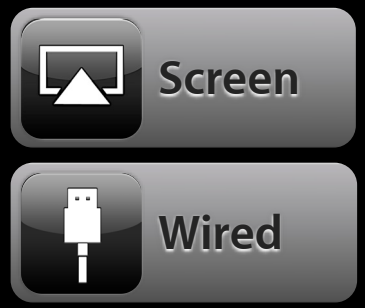
Fullscreen Video Experience

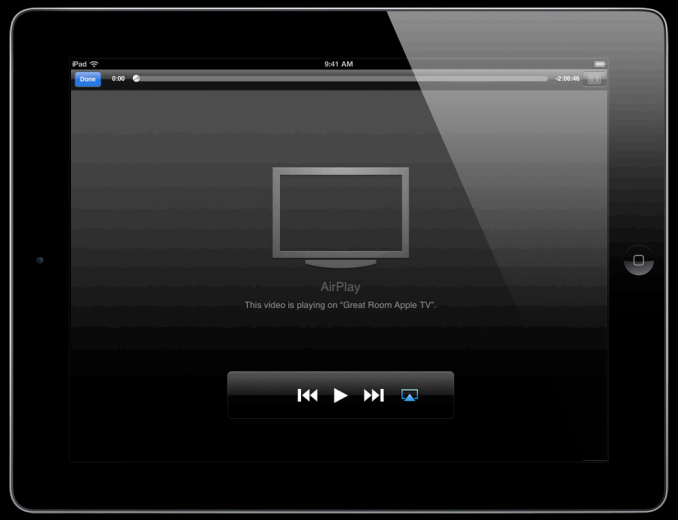
Recommended external screen behavior



Fullscreen Video Experience

Recommended external screen behavior





Fullscreen Video Experience

MPMoviePlayerController

- Automatically takes advantage of external display
- Player needs to be in fullscreen mode

```
[moviePlayer setFullscreen:YES animated:animated];
```

Fullscreen Video Experience

AVPlayer

- Application responsible to take advantage of external display
- When entering fullscreen experience:
  - AirPlay

```
player.usesAirPlayVideoWhileAirPlayScreenIsActive = YES;
```

  - Wired
    - Use `UIScreen` to place video on second display

What You Will Learn

- Making a great AirPlay experience
- Getting the most out of video
- Taking advantage of multiple displays

What You Will Learn

- Making a great AirPlay experience
- Getting the most out of video
- Taking advantage of multiple displays
  - Wired and AirPlay
  - Great for all apps

External Displays

Josh Shaffer

UIScreen

Working with external displays

- Mirroring
- Second display
- Overscan
- Screen modes

http://www.apple.com/
Store
Mac
The amazi
Display a menu
Apple
RSS
Google
iPod
iPhone
iPad
iTunes
Support
Finally.
ng iPhone 4. Now available in white.

iOS Second Display

- Controlled by foreground app
- Distinct display
  - Mirrored display
  - Second display

Mirrored Display

Automatic behavior

Mirrored Display

Automatic behavior

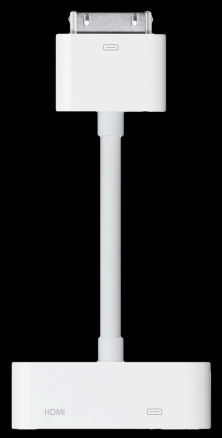
Mirrored Display

Importance of status bar orientation

Mirrored Display

```
[externalScreen mirroredScreen] == [UIScreen mainScreen]
```

- Supported on iPad 2
- Supported with:



Apple Digital AV Adapter

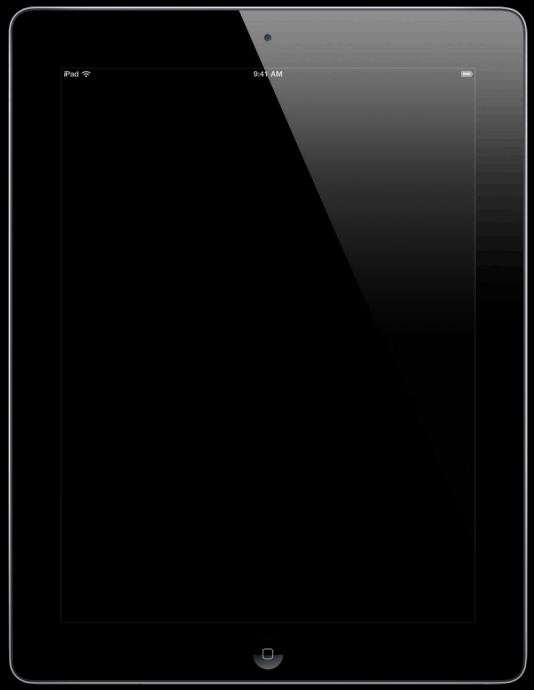
Apple VGA Adapter

Apple TV

Second Display

Second Display

Second Display



Second Display

UIScreen

- Detect an external screen
- Place content on it

Second Display

Detecting at application launch

```
- (void)applicationDidFinishLaunching:(UIApplication *)application
```

Second Display

Detecting at application launch

```
- (void)applicationDidFinishLaunching:(UIApplication *)application
{
    if ([[UIScreen screens] count] > 1) {
```

Second Display

Detecting at application launch

```
- (void)applicationDidFinishLaunching:(UIApplication *)application
{
    if ([[UIScreen screens] count] > 1) {
        [self prepareScreen:[[UIScreen screens] lastObject]];
    }
}
```

Second Display

Detecting display connections

```
[[NSNotificationCenter defaultCenter]
     addObserver:self
        selector:@selector(screenDidConnect:)
            name:UIScreenDidConnectNotification
          object:nil];

- (void)screenDidConnect:(NSNotification *)notification {
    [self prepareScreen:[notification object]];
}
```

Second Display

Displaying content

```
- (void)prepareScreen:(UIScreen *)connectedScreen
{
    CGRect frame = connectedScreen.bounds;
    UIWindow *window = [[UIWindow alloc] initWithFrame:frame];

    [window setScreen:connectedScreen];
    window.hidden = NO;
}
```

Overscan

Overscan

Overscan

Overscan

Compensation

```
@property (nonatomic) UIScreenOverscanCompensation overscanCompensation;
UIScreenOverscanCompensationScale
UIScreenOverscanCompensationInsetBounds
UIScreenOverscanCompensationInsetApplicationFrame
```

Scale (Default)

Generic UIKit content

- Pros
  - 720p framebuffer
  - No additional work
- Cons
  - Lower quality
  - Lower performance

Inset Bounds

High performance UIKit content

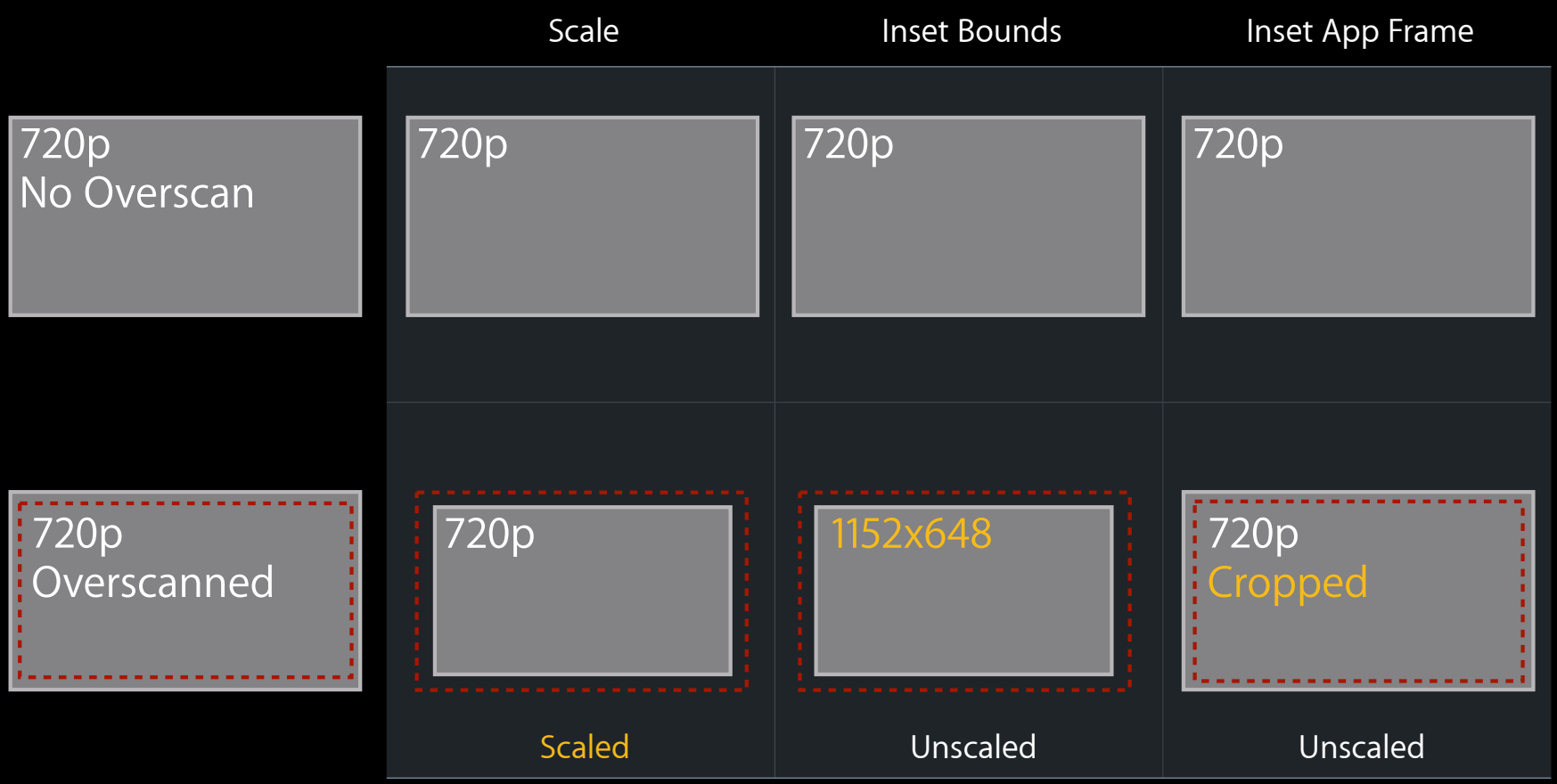
- Pros
  - Smaller screen bounds
  - No scaling
  - Full performance
- Cons
  - <720p framebuffer
  - Resolution independence

Inset Application Frame

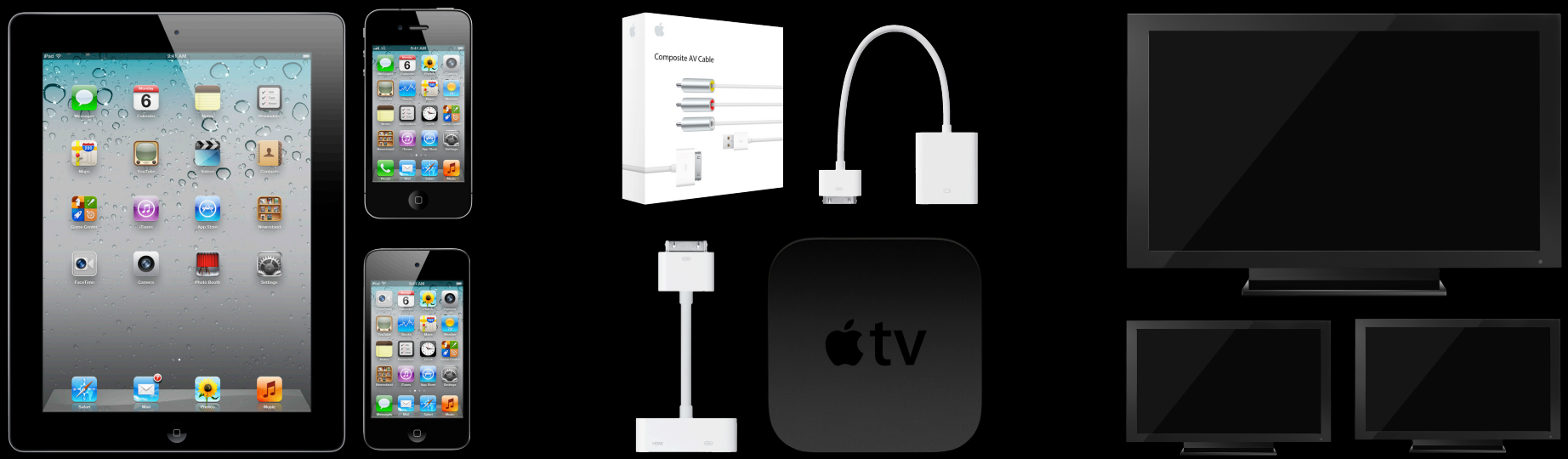
Games and video content

- Pros
  - 720p framebuffer
  - Full screen bounds
  - Smaller application frame
  - No scaling
  - Full performance
- Cons
  - Independent positioning

Overscan

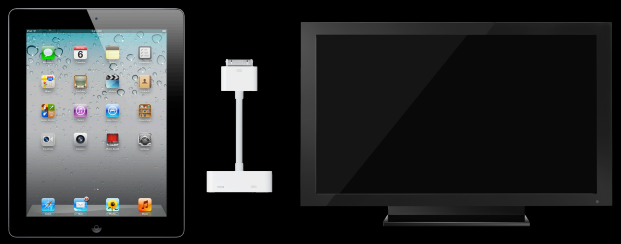

Scale
Inset Bounds
Inset App Frame
720p
No Overscan
720p
720p
720p
720p
Overscanned
720p
1152x648
720p
Cropped
Scaled
Unscaled
Unscaled

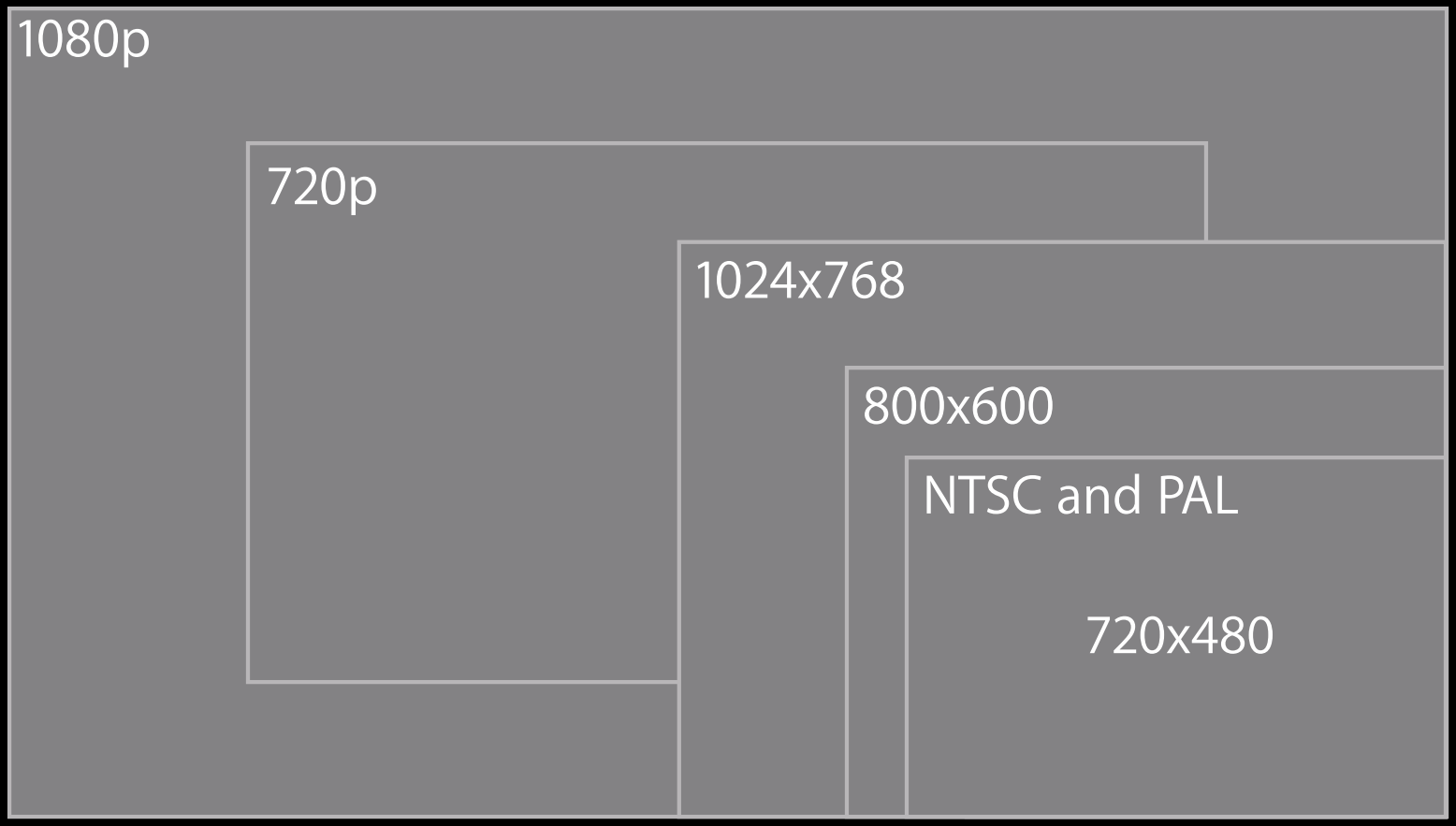
Screen Modes



Screen Modes

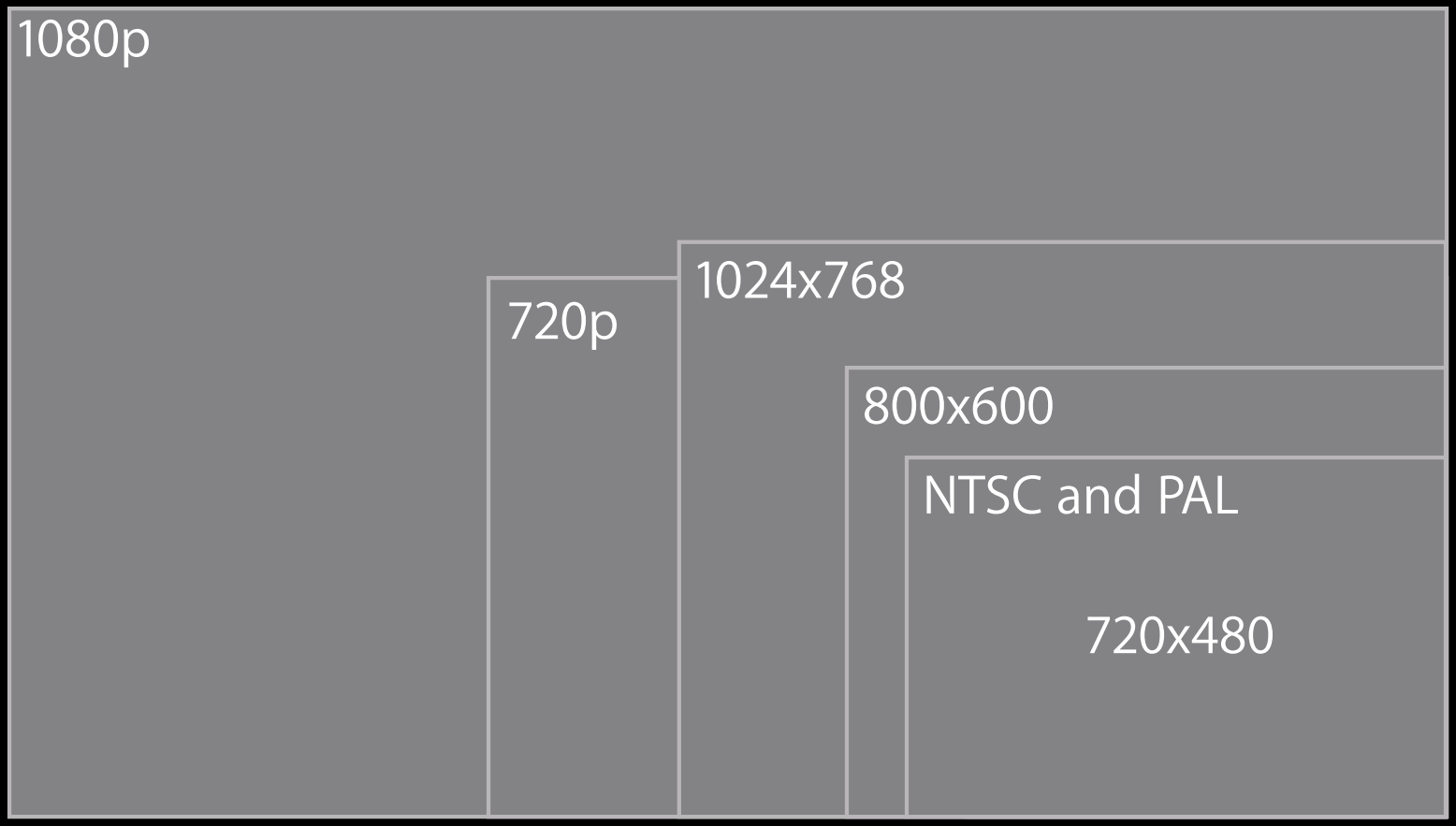
Screen Modes





Screen Modes

Preferred mode

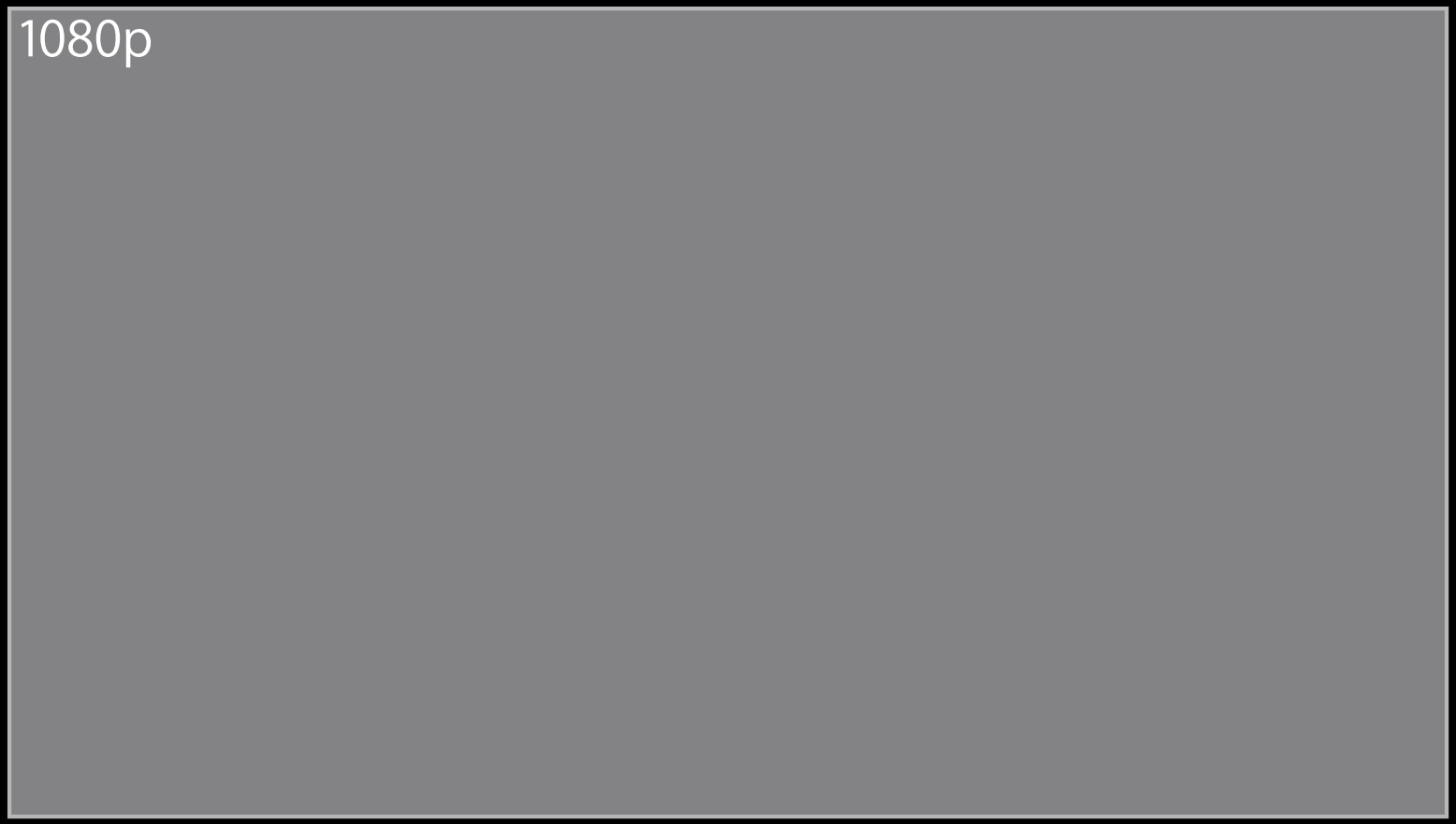


Screen Modes

Preferred mode

Screen Modes

Preferred mode



Screen Modes

Preferred mode

Use the Preferred Mode

Screen Modes

Preferred mode

- Chosen automatically by default
- No hardware mode switch
  - Avoids screen flashes
  - Seamless transition between mirroring
  - Faster launch times
- Best aspect ratio
- Best quality

Screen Mode Performance Tradeoffs

When you cannot render fast enough

Strategy	Advantages	Disadvantages
Preferred Mode **Render full resolution**	Optimal quality Avoids mode switch	Requires more performance tuning Lower performance on larger displays
Preferred Mode **Render lower resolution** **Use layer scaling**		Larger memory use than necessary May introduce scaling artifacts
Choose specific mode	Optimize performance for a common mode	Requires mode switch Requires pass through TV scaler

Do not Ask the User

Summary

- Always set the status bar orientation
- Adjust for overscan
- Use the preferred mode

External Displays in Real Racing 2 HD

Case study

Jacques Philippe Gasselin de Richebourg
Senior Programmer
Firemint

FIREMINT REAL RACING

- First released for iPhone and iPad
- Apple Design Award Winner 2010
- OpenGL ES 1.1 fixed-function rendering pipeline
- 4 car models across 12 tracks

REAL RACING 2 HD

- Updated for iPad 2
- OpenGL ES 2.0 shader rendering pipeline
- Per-pixel lighting, MSAA, reflections
- 10x-20x more geometry
- External display support

Demo

Tips and Tricks

- Gracefully handle display connect and disconnect events
- Choose what to draw on each screen
- Support orientation changes on device
- Test across variety of displays
- Consider impact of screen resolution on performance

Summary

- Explore AirPlay
- Polish your app
- Take advantage of second display

More Information

Allan Shaffer
Graphics and Game Technologies Evangelist
aschaffer@apple.com

Eryk Vershen
Media Technologies Evangelist
evershen@apple.com

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Audio Development for Games	Mission Tuesday 11:30AM
Exploring AV Foundation	Presidio Tuesday 2:00PM

HTTP Live Streaming Update	Nob Hill Tuesday 4:30PM
Advanced HTML5 Media Controllers in Safari	Marina Tuesday 2:00PM
Audio Session Management on iOS	Marina Wednesday 11:30AM
Best Practices for OpenGL ES Apps in iOS	Mission Wednesday 4:30PM

Labs

HTML5 Audio and Video Lab	Internet and Web Lab A Tuesday 4:30PM
AirPlay Lab	Graphics, Media & Games Lab B Wednesday 9:00AM
AV Foundation Lab	Graphics, Media & Games Lab C Wednesday 9:00AM
HTTP Live Streaming Lab	Graphics, Media & Games Lab D Wednesday 9:00AM
Audio Labs	Graphics, Media & Games Lab C Wednesday 2:00PM
AV Foundation Lab	Graphics, Media & Games Lab B Thursday 9:00AM
OpenGL ES Lab	Graphics, Media & Games Lab A Thursday 9:00AM and 2:00PM

WWDC2011

[redacted]

[redacted]

[redacted]